



Subscribe Register Login
(Full Service) (Limited Service, Free)

Search: ☒ The ACM Digital Library ☐ The Guide

extensible <and> macro <and> language

THE ACM DIGITAL LIBRARY

Fe

Terms used extensible and macro and language

Sort results
by

relevance



Display results

expanded form



Save results to a Binder

T

Search Tips

T

☐ Open results in a new window

Results 101 - 120 of 200

Result page: previous 1 2 3 4 5 6

Best 200 shown

101 A retargetable microcode generation system for a high-level microprogra

Peter Marwedel

December 1981

Proceedings of the 14th annual workshop on Microprogran

Full text available: pdf(744.53 KB)

Additional Information: full citation, abstract, reference

A system for the generation of microcode from a high-level microprogrammir independent of the target machine because it is table-driven by a separate h horizontally microprogrammed machines.

102 Expansion-passing style: beyond conventional macros

R. Kent Dybvig, Daniel P. Friedman, Christopher T. Haynes

August 1986 Proceedings of the 1986 ACM conference on LISP and functional pro

Full text available: pdf(566.53 KB)

Additional Information: full citation, references, citings

103 PSAIL: A portable SAIL to C compiler—description and tutorial

P. F. Lemkin

October 1988

ACM SIGPLAN Notices, Volume 23 Issue 10

Full text available: pdf(1.32 MB)

Additional Information: full citation, index terms

104 Contributions: Roster of graphic languages and general subroutine pack

Toby Berk, Arie Kaufman

February 1980

ACM SIGGRAPH Computer Graphics, Volume 13 Issue 4

Full text available: pdf(724.26 KB)


Additional Information: full citation

106 Towards a computational formalization of natural language semantics

Robert M. Schwarcz

September 1969

Proceedings of the 1969 conference on Computational lin

Full text available:  pdf(1.45 MB)

Additional Information: full citation, abstrac

The formalization of natural language semantics is a problem central to a nur concerns. A semantic theory requires a formalized representation of message and the processes of encoding and docoding that relate them. Formal logic h on the notions of model, extension, and intention; with certain changes and a needed for a theory of natural lang ...

106 Programming Languages: GPL, a truly general purpose language

Jan V. Garwick

September 1968

Communications of the ACM, Volume 11 Issue 9

Full text available:  pdf(695.05 KB)

Additional Information: full citation, abstract, re

A truly general purpose programming language, GPL, is described which cont language) new data types as well as facilities for operations performed upon sense that no basic element can be derived from the others with high efficier the ALGOL 60 for-statements, and if-statements are not basic; they are spec “symbols” (underline ...

Keywords: ALGOL, general purpose, macro, programming language, self-exte

107 SLX: pyramid power

James O. Henriksen

December 1999 Proceedings of the 31st conference on Winter simulation: Simulat

Full text available:  pdf(116.24 KB)

Additional Information: full citation, references, cit

108 A programmer controlled approach to data and control abstraction

Juha Heinänen

June 1983 Proceedings of the 1983 ACM SIGPLAN symposium on Programming l

Full text available:  pdf(1.02 MB)

Additional Information: full citation, abstract, referen

Traditionally, data abstraction languages have only provided a means to exte include new procedures and data types not present in the base language. Thi approach, which also allows programmers to extend the language “do of the previously preempted decisions concerning the nature and implementa order to illustrate the approach, several e ...

109 Toward a formal theory of extensible software

Shriram Krishnamurthi, Matthias Felleisen

November 1998 ACM SIGSOFT Software Engineering Notes , Proceedings of the 6th
on Foundations of software engineering, Volume 23 Issue 6

Full text available:  pdf(862.50 KB)


Additional Information: full citation, abstract, reference

As software projects continue to grow in scale and scope, it becomes important to reuse is *extensibility*, i.e., the extension of software without accessing existing code. We propose a rigorous, semantics-based definition of software extensibility. Then we apply them to several programs. The examination shows how programming can be done in the ...

110 Type-safe linking and modular assembly language

Neal Glew, Greg Morrisett

January 1999 Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of Programming Languages



Full text available:  pdf(1.36 MB)

Additional Information: full citation, references, citations, abstract

111 DATR: a language for lexical knowledge representation

Roger Evans, Gerald Gazdar

June 1996 Computational Linguistics, Volume 22 Issue 2

Full text available:  pdf(3.14 MB)  Publisher Site

Additional Information: full citation, abstract, reference

Much recent research on the design of natural language lexicons has made use of structures originally developed for general knowledge representation purposes in Artificial Intelligence. We propose a language for defining nonmonotonic inheritance networks with path/value equations specifically for lexical knowledge representation. In keeping with its intended use, the language constructs embodied ...

112 A compiler language for data structures

Neal Laurance

January 1968 Proceedings of the 1968 23rd ACM national conference

Full text available:  pdf(727.45 KB)

Additional Information: full citation, abstract, reference

The subject of data structures has received a great deal of attention in the past few years. In computer-aided design. Programming systems used for creating data structures (e.g., "graphical languages") vary greatly in the rigidity of their representation to the programmer. As an example of a high-level system, we can mention the ...

110 Template meta-programming for Haskell

Tim Sheard, Simon Peyton Jones

October 2002

Proceedings of the ACM SIGPLAN workshop on Haskell

Full text available:  pdf(169.20 KB)

Additional Information: full citation, abstract, referenc

We propose a new extension to the purely functional programming language *meta-programming*. The purpose of the system is to support the *algorithmic* compile-time. The ability to generate code at compile time allows the program polytypic programs, macro-like expansion, user directed optimization (such a data structures and functions from existing ...

Keywords: meta programming, templates

114 PLI workshops: Template meta-programming for Haskell

Tim Sheard, Simon Peyton Jones

December 2002

ACM SIGPLAN Notices, Volume 37 Issue 12

Full text available:  pdf(244.61 KB)

Additional Information: full citation, abstract, refe

We propose a new extension to the purely functional programming language *meta-programming*. The purpose of the system is to support the *algorithmic* compile-time. The ability to generate code at compile time allows the program polytypic programs, macro-like expansion, user directed optimization (such a data structures and functions from existing ...

Keywords: Meta programming, templates

115 FORTRAN IV as a syntax language

B. M. Leavenworth

February 1964

Communications of the ACM, Volume 7 Issue 2

Full text available:  pdf(921.80 KB) Additional Information: full citation, references, citings, index terms

116 Revised report on the algorithmic language scheme

J Rees, W Clinger

December 1986

ACM SIGPLAN Notices, Volume 21 Issue 12

Full text available:  pdf(4.06 MB) Additional Information: full citation, citings, index terms

117 An extensible interpreter

Stephen A. Schuman

September 1971 ACM SIGPLAN Notices , Proceedings of the international symposium
Issue 12

Full text available:  pdf(595.31 KB)


Additional Information: full citation, abstract, c

The purpose of this brief paper is to propose an alternative approach for the
The idea outlined here will be referred to as an extensible interpreter. In this
an exceptionally concise description of the basic concept, even though there
the same notion, namely as a strategy for integrating an optimizing compiler

118 Josh: an open AspectJ-like language

Shigeru Chiba, Kiyoshi Nakagawa

March 2004 Proceedings of the 3rd international conference on Aspect-oriented

Full text available:  pdf(1.09 MB)

Additional Information: full citation, abstract,


Although aspect-oriented programming (AOP) is becoming widely used, the c
generic and reusable description of advice are still research topics. To addres
which is our new AspectJ-like language with an extensible pointcut language
description. The extensible pointcut language is based on the idea of open co
pointcut designator in Java, the ...

Keywords: extensibility, generic description, pointcut

119 M-LISP: a representation-independent dialect of LISP with reduction sen

Robert Muller

October 1992 ACM Transactions on Programming Languages and Systems (TOPLAS)

Full text available:  pdf(1.67 MB)

Additional Information: full citation, abstract, references, c

In this paper we introduce M-LISP, a dialect of LISP designed with an eye to
with the structural style of operational semantics advocated by Plotkin [28].
of LISP [20] in an attempt to clarify the source of its metalinguistic power. W
clause in this definition. We then define the abstract syntax and operational s

Keywords: fexprs, metalinguistic constructs, reflection, reification, unquote

120 Some prolog macros for rule-based programming: why? how?

Tim Menzies, Lindsay Mason

October 2002 Proceedings of the 2002 ACM SIGPLAN workshop on Rule-based




Full text available:  pdf(530.35 KB)

Additional Information: full citation, abstract, refer

The history, benefits, and drawbacks to pure rule-based programming is disc
rule-based programming is described. The extensions are very quick to code
range of knowledge engineering applications.

Keywords: history, prolog, rule-based programming

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Co](#)

Useful downloads:  Adobe Acrobat  QuickTime  Windows M



Subscribe Register Login
(Full Service) (Limited Service, Free)

Search: ☒ The ACM Digital Library ☐ The Guide

extensible <and> macro <and> language

THE ACM DIGITAL LIBRARY

Fe

Terms used extensible and macro and language

Sort results
by

relevance

☒ Save results to a Binder

T

☒ Search Tips

T

☐ Open results in a new window

Display results

expanded form

Results 1 - 20 of 200

Result page: 1 2 3 4 5 6 7 8 9

Best 200 shown

1 A graded bibliography on macro systems and extensible languages

John R. Metzner

January 1979

ACM SIGPLAN Notices, Volume 14 Issue 1

Full text available: pdf(830.04 KB)

Additional Information: full citation, references

2 Macro instruction extensions of compiler languages

M. Douglas McIlroy

April 1960

Communications of the ACM, Volume 3 Issue 4

Full text available: pdf(831.69 KB)

Additional Information: full citation, abstract, ref

Macroinstruction compilers constructed from a small set of functions can be m conditional assembly, nested definitions, and parenthetical notation serve to n general extensions to its ground language.

3 Macro processing in high-level languages

Alexander Sakharov

November 1992

ACM SIGPLAN Notices, Volume 27 Issue 11

Full text available: pdf(709.71 KB)

Additional Information: full citation, abstra

A macro language is proposed. It enables macro processing in high-level prog this language refer to the grammars of the respective programming language; programming languages. It is described how to automatically generate macro programming language grammars written in the lex-yacc format. Examples of of macros are given.

4 A model of extensible language systems

M. G. Notley

September 1971 ACM SIGPLAN Notices , Proceedings of the international symposium
12

Full text available:  pdf(333.35 KB)

Additional Information: full citation, abstract, ci

At the present time the subject of extensible languages appears to suffer from
to knit together the many pieces of individual work that are being done. This p
lack. What appears to be required is some central conceptual model or paradig
onto which framework we can hang all these pieces of work. This paper, there

5 Growing languages with metamorphic syntax macros

Claus Brabrand, Michael I. Schwartzbach

January 2002 ACM SIGPLAN Notices , Proceedings of the 2002 ACM SIGPLAN work
semantics-based program manipulation, Volume 37 Issue 3

Full text available:  pdf(217.81 KB)

Additional Information: full citation, abstract, re


*"From now on, a main goal in designing a language should be to plan for grow
OOPSLA '98 invited talk. We present our experiences with a syntax macro lang
abstraction mechanism for growing (domain-specific) extensions of programm
is designed to guarantee type safety and termination. A concept of metamorph
inductively def ...*

6 Experience with an extensible language

Edgar T. Irons

January 1970

Communications of the ACM, Volume 13 Issue 1

Full text available:  pdf(1.17 MB)

Additional Information: full citation, abstract, refer

An operational extensible language system is described. The system and its b
efficiency, flexibility, and utility for different categories of users.

Keywords: ambiguity, bootstrapping, compiler, extensible, programming langu

7 Maya: multiple-dispatch syntax extension in Java

Jason Baker, Wilson C. Hsieh

May 2002 ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Confere
implementation, Volume 37 Issue 5

Full text available:  pdf(152.75 KB)

Additional Information: full citation, abstract, reference

We have designed and implemented Maya, a version of Java that allows progr
syntax. Maya generalizes macro systems by treating grammar productions as
productions as multimethods on the corresponding generic functions. Program
grammar productions) and new multimethods (i.e., semantic actions), through
language and change the semantics of ...

Keywords: Java, generative programming, macros, metaprogramming

8 Stack Machines and Classes of Nonnested Macro Languages

Joost Engelfriet, Erik Meineche Schmidt, Jan van Leeuwen

January 1980 Journal of the ACM (JACM), Volume 27 Issue 1

Full text available:  pdf(1.46 MB) Additional Information: full citation, references, citings, index terms

9 A language independent macro processor

William M. Waite

July 1967 Communications of the ACM, Volume 10 Issue 7


Full text available:  pdf(1.06 MB) Additional Information: full citation, abstract, references

The problem of obtaining starting values for the Newton-Raphson calculation is considered. It is shown that the conventionally used best uniform approximation starting values. The problem of obtaining optimal starting values is stated, and optimal polynomial starting values is given.

10 Hygienic macro expansion

Eugene Kohlbecker, Daniel P. Friedman, Matthias Felleisen, Bruce Duba

August 1986 Proceedings of the 1986 ACM conference on LISP and functional programming

Full text available:  pdf(762.23 KB) Additional Information: full citation, references, citings

11 A brief look at extension programming before and now

Liisa R ih a

February 1995 ACM SIGPLAN Notices, Volume 30 Issue 2

Full text available:  pdf(898.76 KB) Additional Information: full citation, abstract

We try to bind together some old and some new: what is an extension. In addition, we discuss facilities in three language systems with slightly different theoretical basis. We conclude with Language, with additional comments to e.g., C++.

12 Technical contributions: STRCMACS: an extensive set of macros to aid in assembly language

C. Wrandle Barth

August 1976 ACM SIGPLAN Notices, Volume 11 Issue 8

Full text available:  pdf(219.18 KB) Additional Information:

13 Composable and compilable macros: you want it when?

Matthew Flatt

September 2002 ACM SIGPLAN Notices , Proceedings of the seventh ACM SIGPLAN programming, Volume 37 Issue 9

Full text available:  pdf(162.46 KB)

Additional Information: full citation, abstract, reference

Many macro systems, especially for Lisp and Scheme, allow macro transformers. Moreover, the language for implementing compile-time macro transformers is implementing run-time functions. As a side effect of this sharing, implementing compile-time values and run-time values, as well as values from separate compiler programming tools that must parse code without executing it ...

Keywords: language tower, macros, modules

14 Macros as multi-stage computations: type-safe, generative, binding macros

Steven E. Ganz, Amr Sabry, Walid Taha

October 2001 ACM SIGPLAN Notices , Proceedings of the sixth ACM SIGPLAN international programming, Volume 36 Issue 10

Full text available:  pdf(233.27 KB)


Additional Information: full citation, abstract, reference

With few exceptions, macros have traditionally been viewed as operations on source code. This view makes macros seem ad hoc, and is at odds with two desirable features of macros: static typing and static scoping. At a deeper level, there is a need for a simple language that argues that these problems can be addressed by formally viewing macros as computations. This paper eliminates the need for fresh identifiers ...

15 A history of the SNOBOL programming languages

Ralph E. Griswold

January 1978 ACM SIGPLAN Notices , The first ACM SIGPLAN conference on History of programming, Issue 8

Full text available:  pdf(3.56 MB)

Additional Information: full citation, abstract, reference

Development of the SNOBOL language began in 1962. It was followed by SNOBOL1, SNOBOL2 and SNOBOL3 (which were closely related), the others differ substantially. They are considered separate languages than versions of one language. In this paper history of the language, SNOBOL, although important aspects of the subsequent languages ...

16 Extensible languages: A potential user's point of view

J. J. Duby

September 1971 ACM SIGPLAN Notices , Proceedings of the international symposium on programming, 12

Full text available:  pdf(215.82 KB)

Additional Information: full citation, abstract, reference

The purpose of this paper is to describe a computer user's concern about what a language should be like and how they will do it. It may in some respects sound a little demagogic, by considering requirements. However, as Lenin said, "facts are stubborn", and performance characteristics of usual programming languages, they will want to use extensible languages, and in fact ...

17 Macros in higher-level languages

Henry J. Bowlden

September 1971 ACM SIGPLAN Notices , Proceedings of the international symposium
12

Full text available:  pdf(239.75 KB)

Additional Information: full citation, abstract, references

The concept of macros, as a tool to extend the expressive capability of a symbol substitution, has existed in the lexicon of programming languages for many years in connection with assemblers; in some cases capabilities not too far short of the "higher-level languages" have been provided by this means. Att similar tools have, however, not be ...

18 Extensible data features in the operating system language OSL/2

Peter A. Alsberg

October 1971 Proceedings of the third ACM symposium on Operating systems

Full text available:  pdf(482.55 KB)

Additional Information: full citation, abstract, references

The extensible data facilities of OSL/2, an operating system language, are described. New data types, such as queues, files, and tables, and describe complex access methods used in operating system codes, data type extension facilities help the programmer with complex data manipulation and provide logical places to insert and remove symbols. Implementation of these facilities is ...

19 Technical contributions: Experience with extensible, portable Fortran extensions

A. James Cook

September 1976 ACM SIGPLAN Notices, Volume 11 Issue 9

Full text available:  pdf(497.43 KB)

Additional Information: full citation, abstract, references

We assess the impact over a three-year period, of the macro-pre-processor MPP on Fortran processes. We confine our assessment to SLAC and Stanford since, although in the United States and to a lesser extent in Europe, we have no personal knowledge attributed to three factors: (1) portability, (2) compatibility (with existing FORTRAN) is sub-divided into (a) extensions ...

20 MACRO: a programming language

Stephen R. Greenwood

December 1979 ACM SIGPLAN Notices, Volume 14 Issue 12

Full text available:  pdf(1.41 MB) Additional Information: full citation, references, citations

Results 1 - 20 of 200

Result page: **1** 2 3 4 5 6

The ACM Portal is published by the Association for Computing Machinery

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  Adobe Acrobat  QuickTime  Windows Media



Subscribe Register Login
(Full Service) (Limited Service, Free)

Search: ☒ The ACM Digital Library ☐ The Guide

extensible <and> macro <and> language

THE ACM DIGITAL LIBRARY

Fe

Terms used extensible and macro and language

Sort results
by

relevance

☒ Save results to a Binder

☒ Search Tips

☐ Open results in a new window

Display results

expanded form

Results 1 - 20 of 200

Result page: 1 2 3 4 5 6 7 8 9

Best 200 shown

1 A graded bibliography on macro systems and extensible languages

John R. Metzner

January 1979

ACM SIGPLAN Notices, Volume 14 Issue 1

Full text available: pdf(830.04 KB)

Additional Information: full citation, references

2 Macro instruction extensions of compiler languages

M. Douglas McIlroy

April 1960

Communications of the ACM, Volume 3 Issue 4

Full text available: pdf(831.69 KB)

Additional Information: full citation, abstract, ref

Macroinstruction compilers constructed from a small set of functions can be m conditional assembly, nested definitions, and parenthetical notation serve to n general extensions to its ground language.

3 Macro processing in high-level languages

Alexander Sakharov

November 1992

ACM SIGPLAN Notices, Volume 27 Issue 11

Full text available: pdf(709.71 KB)

Additional Information: full citation, abstra

A macro language is proposed. It enables macro processing in high-level prog this language refer to the grammars of the respective programming language: programming languages. It is described how to automatically generate macro programming language grammars written in the lex-yacc format. Examples of of macros are given.

4 A model of extensible language systems

M. G. Notley

September 1971 ACM SIGPLAN Notices , Proceedings of the international symposium
12

Full text available:  pdf(333.35 KB)

Additional Information: full citation, abstract, ci

At the present time the subject of extensible languages appears to suffer from
to knit together the many pieces of individual work that are being done. This p
lack. What appears to be required is some central conceptual model or paradig
onto which framework we can hang all these pieces of work. This paper, there

5 Growing languages with metamorphic syntax macros

Claus Brabrand, Michael I. Schwartzbach

January 2002 ACM SIGPLAN Notices , Proceedings of the 2002 ACM SIGPLAN work
semantics-based program manipulation, Volume 37 Issue 3

Full text available:  pdf(217.81 KB)

Additional Information: full citation, abstract, re

*"From now on, a main goal in designing a language should be to plan for grow
OOPSLA '98 invited talk. We present our experiences with a syntax macro lang
abstraction mechanism for growing (domain-specific) extensions of programm
is designed to guarantee type safety and termination. A concept of metamorph
inductively def ...*

6 Experience with an extensible language

Edgar T. Irons

January 1970 Communications of the ACM, Volume 13 Issue 1

Full text available:  pdf(1.17 MB)

Additional Information: full citation, abstract, refer

An operational extensible language system is described. The system and its b
efficiency, flexibility, and utility for different categories of users.

Keywords: ambiguity, bootstrapping, compiler, extensible, programming langu

7 Maya: multiple-dispatch syntax extension in Java

Jason Baker, Wilson C. Hsieh

May 2002 ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Confere
implementation, Volume 37 Issue 5

Full text available:  pdf(152.75 KB)

Additional Information: full citation, abstract, reference

We have designed and implemented Maya, a version of Java that allows progr
syntax. Maya generalizes macro systems by treating grammar productions as
productions as multimethods on the corresponding generic functions. Program
grammar productions) and new multimethods (i.e., semantic actions), through
language and change the semantics of ...

Keywords: Java, generative programming, macros, metaprogramming

9 Stack Machines and Classes of Nonnested Macro Languages

Joost Engelfriet, Erik Meineche Schmidt, Jan van Leeuwen


January 1980 Journal of the ACM (JACM), Volume 27 Issue 1

Full text available:  pdf(1.46 MB) Additional Information: full citation, references, citings, index terms

9 A language independent macro processor

William M. Waite

July 1967 Communications of the ACM, Volume 10 Issue 7

Full text available:  pdf(1.06 MB) Additional Information: full citation, abstract, references

The problem of obtaining starting values for the Newton-Raphson calculation is considered. It is shown that the conventionally used best uniform approximation starting values. The problem of obtaining optimal starting values is stated, and optimal polynomial starting values is given.

10 Hygienic macro expansion

Eugene Kohlbecker, Daniel P. Friedman, Matthias Felleisen, Bruce Duba

August 1986 Proceedings of the 1986 ACM conference on LISP and functional programming

Full text available:  pdf(762.23 KB) Additional Information: full citation, references, citings

11 A brief look at extension programming before and now

Liisa Räihä

February 1995 ACM SIGPLAN Notices, Volume 30 Issue 2

Full text available:  pdf(898.76 KB) Additional Information: full citation, abstract

We try to bind together some old and some new: what is an extension. In addition, we discuss facilities in three language systems with slightly different theoretical basis. We conclude with Language, with additional comments to e.g., C++.

12 Technical contributions: STRCMACS: an extensive set of macros to aid in assembly language

C. Wrangle Barth

August 1976 ACM SIGPLAN Notices, Volume 11 Issue 8

Full text available:  pdf(219.18 KB) Additional Information:

13 Composable and compilable macros:.. you want it when?

Matthew Flatt

September 2002 ACM SIGPLAN Notices , Proceedings of the seventh ACM SIGPLAN programming, Volume 37 Issue 9

Full text available:  pdf(162.46 KB)

Additional Information: full citation, abstract, reference

Many macro systems, especially for Lisp and Scheme, allow macro transformers. Moreover, the language for implementing compile-time macro transformers is implementing run-time functions. As a side effect of this sharing, implementing compile-time values and run-time values, as well as values from separate compilation tools that must parse code without executing it ...

Keywords: language tower, macros, modules

14 Macros as multi-stage computations: type-safe, generative, binding macros

Steven E. Ganz, Amr Sabry, Walid Taha

October 2001 ACM SIGPLAN Notices , Proceedings of the sixth ACM SIGPLAN international programming, Volume 36 Issue 10

Full text available:  pdf(233.27 KB)


Additional Information: full citation, abstract, reference

With few exceptions, macros have traditionally been viewed as operations on source code. This view makes macros seem ad hoc, and is at odds with two desirable features of macros: static typing and static scoping. At a deeper level, there is a need for a simple language that argues that these problems can be addressed by formally viewing macros as computations. This paper eliminates the need for fresh identifiers ...

15 A history of the SNOBOL programming languages

Ralph E. Griswold

January 1978 ACM SIGPLAN Notices , The first ACM SIGPLAN conference on History of programming, Issue 8

Full text available:  pdf(3.56 MB)

Additional Information: full citation, abstract, reference

Development of the SNOBOL language began in 1962. It was followed by SNOBOL1, SNOBOL2 and SNOBOL3 (which were closely related), the others differ substantially. They are considered separate languages than versions of one language. In this paper history of the language, SNOBOL, although important aspects of the subsequent languages ...

16 Extensible languages: A potential user's point of view

J. J. Duby

September 1971 ACM SIGPLAN Notices , Proceedings of the international symposium on programming, 12

Full text available:  pdf(215.82 KB)

Additional Information: full citation, abstract, reference

The purpose of this paper is to describe a computer user's concern about what languages will do it. It may in some respects sound a little demagogic, by considering requirements. However, as Lenin said, "facts are stubborn", and performance characteristics of usual programming languages, they will want to use extensible languages, and in fact ...

17 Macros in higher-level languages

Henry J. Bowlden

September 1971 ACM SIGPLAN Notices , Proceedings of the international symposium
12

Full text available:  pdf(239.75 KB)

Additional Information: full citation, abstract, references

The concept of macros, as a tool to extend the expressive capability of a symbolic substitution, has existed in the lexicon of programming languages for many years. In connection with assemblers; in some cases capabilities not too far short of the "higher-level languages" have been provided by this means. At the same time, similar tools have, however, not been ...

18 Extensible data features in the operating system language OSL/2

Peter A. Alsberg

October 1971 Proceedings of the third ACM symposium on Operating systems

Full text available:  pdf(482.55 KB)

Additional Information: full citation, abstract, references

The extensible data facilities of OSL/2, an operating system language, are described. New data types, such as queues, files, and tables, and describe complex access methods used in operating system codes, data type extension facilities help the programmer. Complex data manipulation and provide logical places to insert and remove symbols. Implementation of these facilities is ...

19 Technical contributions: Experience with extensible, portable Fortran extensions

A. James Cook

September 1976

ACM SIGPLAN Notices, Volume 11 Issue 9

Full text available:  pdf(497.43 KB)

Additional Information: full citation, abstract, references

We assess the impact over a three-year period, of the macro-pre-processor MPP on Fortran processes. We confine our assessment to SLAC and Stanford since, although MPP is used in the United States and to a lesser extent in Europe, we have no personal knowledge attributed to three factors: (1) portability, (2) compatibility (with existing FORTRAN) is sub-divided into (a) extensions ...

20 MACRO: a programming language

Stephen R. Greenwood

December 1979 ACM SIGPLAN Notices, Volume 14 Issue 12

Full text available:  pdf(1.41 MB) Additional Information: full citation, references, citations

Results 1 - 20 of 200

Result page: **1** 2 3 4 5 6

The ACM Portal is published by the Association for Computing Machinery

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  Adobe Acrobat  QuickTime  Windows Media

Welcome to IEEE Xplore®

- ☐ [Home](#)
- ☐ [What Can I Access?](#)
- ☐ [Log-out](#)

Tables of Contents

- ☐ [Journals & Magazines](#)
- ☐ [Conference Proceedings](#)
- ☐ [Standards](#)

Search

- ☐ [By Author](#)
- ☐ [Basic](#)
- ☐ [Advanced](#)

Member Services

- ☐ [Join IEEE](#)
- ☐ [Establish IEEE Web Account](#)
- ☐ [Access the IEEE Member Digital Library](#)

IEEE Enterprise

- ☐ [Access the IEEE Enterprise File Cabinet](#)

Your search matched **0** of **1051129** documents.

A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:

You may refine your search by editing the current search expression or enter a new one in the text box.

☐ Check to search within this result set

Results Key:

JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

Results:

No documents matched your query.



[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Web Account](#) | [New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#) | [No Robots Please](#) | [Release Notes](#) | [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2004 IEEE — All rights reserved

Welcome to IEEE Xplore®

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

IEEE Enterprise

- ☐ Access the IEEE Enterprise File Cabinet



Print Format

Your search matched **8** of **1051129** documents.

A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:

You may refine your search by editing the current search expression or enter a new one in the text box.

☐ Check to search within this result set

Results Key:

JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 The Camelot library: A C language extension for programming a general purpose distributed transaction system

Bloch, J.J.;

Distributed Computing Systems, 1989., 9th International Conference on , 5-9 1989

Pages:172 - 180

[\[Abstract\]](#) [\[PDF Full-Text \(792 KB\)\]](#) **IEEE CNF**

2 Shared-memory parallel programming in C++

Beck, B.;

Software, IEEE , Volume: 7 , Issue: 4 , July 1990

Pages:38 - 48

[\[Abstract\]](#) [\[PDF Full-Text \(1088 KB\)\]](#) **IEEE JNL**

3 GUI approach to programming of TMO frames

Kim, K.H.; Seok-Joong Kang; Yuqing Li;

Object-Oriented Real-Time Dependable Systems, 2002. (WORDS 2002). Proceedings of the Seventh International Workshop on , 7-9 Jan. 2002

Pages:19 - 26

[\[Abstract\]](#) [\[PDF Full-Text \(491 KB\)\]](#) **IEEE CNF**

4 HyperSnapping

Masui, T.;

Human-Centric Computing Languages and Environments, 2001. Proceedings Symposia on , 5-7 Sept. 2001

Pages:188 - 194

[\[Abstract\]](#) [\[PDF Full-Text \(430 KB\)\]](#) **IEEE CNF**

5 Development strategies for durable hardware language descriptions analog and mixed signal circuits

Murphy, E.; Bibyk, S.;

Circuits and Systems, 2001. MWSCAS 2001. Proceedings of the 44th IEEE 2001

[\[Abstract\]](#) [\[PDF Full-Text \(346 KB\)\]](#) IEEE CNF**6 The COCO^{DEF} approach to COCOLOG logic control***Martinez-Mascarua, C.; Caines, P.E.;*

Decision and Control, 1998. Proceedings of the 37th IEEE Conference on , Vol 4 , 16-18 Dec. 1998

Pages:3783 - 3788 vol.4

[\[Abstract\]](#) [\[PDF Full-Text \(540 KB\)\]](#) IEEE CNF**7 Automated validation of operational flight programs (OFPs) and flig training simulators***Van Fleet, J.; Flannery, S.; Rura, J.;*

Aerospace and Electronics Conference, 1994. NAECON 1994., Proceedings of t IEEE 1994 National , 23-27 May 1994

Pages:1006 - 1013 vol.2

[\[Abstract\]](#) [\[PDF Full-Text \(620 KB\)\]](#) IEEE CNF**8 A microprocessor in 4 months. Development of the FHOP-microprocessor-kernel via VHDL and logic-synthesis***Jansen, D.; Gieringer, T.; Zimpfer, F.;*

ASIC Conference and Exhibit, 1994. Proceedings., Seventh Annual IEEE International , 19-23 Sept. 1994

Pages:79 - 85

[\[Abstract\]](#) [\[PDF Full-Text \(500 KB\)\]](#) IEEE CNF

[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Web Account](#) | [New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#) | [No Robots Please](#) | [Release Notes](#)
| [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2004 IEEE — All rights reserved